Efficient empty seat finding technology proposal using object detection: Empty Seat?

Doyeop Kim [1], Jaeyoon Park [2], Dayeon Woo [3], and Jimin Choi[4]

[1] Sungkyunkwan University, Department of Advanced Materials Science & Engineering
[2] Sungkyunkwan University, School of Electronic and Electrical Engineering
[3] Sungkyunkwan University, Department of Data Science
[4] Sungkyunkwan University, Department of Culture and Technology

Abstract. This document proposes a service that offers real-time seat availability information for study spaces within Sungkyunkwan University's Natural Science Campus. The primary objective of the service is to provide convenience to both students and administrators by delivering real-time seat availability information for on-campus study spaces, thereby maximizing space utilization. The service utilizes the object detection model YOLOv5 to analyze seat availability from real-time images and presents this information to users in a user-friendly UI format. This service aims to address the limitations of existing services and assist users in enjoying a more convenient school life.

Keywords: Object Detection, YOLOv5, Real-time Server, Real-time seat availability, Study spaces

1 Introduction

At Sungkyunkwan University's Natural Science Campus, there are numerous study spaces available. However, based on experiences, Sungkyunkwan University students have found some inconvenience in using these spaces. Real-time seat availability checking is not possible, making it difficult to anticipate how crowded each space is in advance. Additionally, due to the large campus size, finding a suitable study space can consume a large amount of time and energy. Despite investigating campus services such as the Haedong Library Reservation System, and SKKU CAMPUSMAP, it was insufficient in addressing these difficulties. These services were limited to specific areas or could not provide real-time seat availability information.

Our team believes that "Empty Seats?", which combines the computer vision model YOLOv5 with a user-friendly UI, can address these issues. "Empty Seats?" is a service that allows users to check real-time seat availability in various campus spaces. The operation of the service is as follows: First, it receives real-time images from cameras installed in each space. Next, YOLOv5 is used to detect whether there are individuals seated in each seat in the photo or if objects are placed on the seat. Then, the service determines whether each seat is vacant or in use and visually processes the results to provide users with information on seat availability. Through "Empty Seats?", users can easily assess the availability of empty seats in campus spaces without the need to move around.

2 Motivation

Difficulty in accurately determining the availability of learning spaces leads to various side effects.

Firstly, it results in inconvenience for users, primarily students. Presently, there are students who wander around the campus in search of an available seat until they find one. Some students even choose to leave school immediately without attempting to find a seat in the first place, because they are not sure if there are seats. This requires students to spend extra time and energy to find the right study space.

Secondly, it inconveniences space administrators, as they have to devote more effort to ensure that many students can utilize the space conveniently.

Lastly, it adversely affects space utilization. Securing additional study spaces within the campus is subject to various constraints, including financial considerations. Therefore, it becomes imperative to enhance the effective utilization of existing spaces.

In such circumstances, "Empty Seats?" is poised to become the optimal service that provides convenience to both users and administrators while simultaneously enhancing the utilization of existing spaces.

3 Problem Statement

The problems encountered finding an empty seat are as following:

3.1 Difficulty in finding available seats

To check available seats on campus, students must visit all the facilities and spaces on foot. SungKyunKwan University campus, both Seoul and Suwon, is known with its size. In addition, the spaces are not gathered in a single area rather spread apart among the campus. For students, it is easy to encounter unavailable spaces when visited, especially for the ones that are named. Therefore, it requires students an amount of walk to find if one fails in attempt, leading to exhaustion before seating.

3.2 Difficulty in finding real-time information

Even with the reservation system that already exists for some spaces, such as the Central/Samsung Library, there are cases where the information given in service is different from real. When visited, one might be using the seat without reservation. In the opposite, it is easy to encounter seats booked but not used. Although the actual blame is on the students not providing the service with the correct information (through not changing the status of the seat), however it is the students searching for empty seats that face confusion. Therefore, real-time information may increase convenience for students, providing accuracy.

3.3 Difficulty in finding information in single platform

There are platforms that provide information about available facilities and spaces for students to use. However, each contains different types of information. Furthermore, since the number of spaces is not small, one must check the platforms respectively. This process may be inconvenient and troublesome for some. Therefore, a platform that includes information of all spaces would provide convenience to students searching for space to use.

4 Related Work

4.1 SKKU Central/Samsung Library

Fig. 1. Samsung Library Reservation Application

SKKU Central/Samsung Library are main libraries in each Seoul and Suwon campus students use daily. For each campus, there exists a reservation system which allows students to check available seats without visiting themselves. It supports both website and mobile application for both Android and iOS devices, letting students available to use on any device. Through the system, students are available to check seats on all the sections, book them if empty and reserve separate seminar rooms if needed. All students must enter and exit the library with a student ID card, and the library checks if one is present in real-time, deciding to free booked seats depending on the time absent.

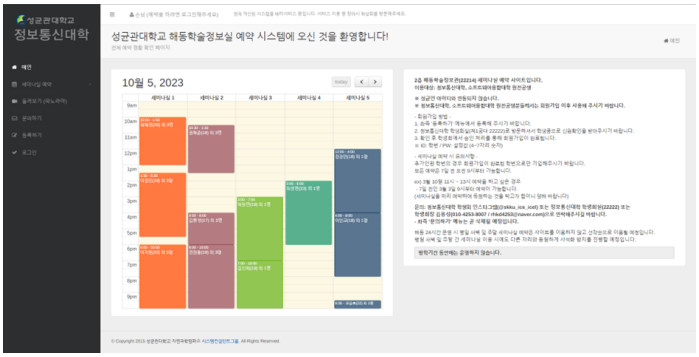

Fig. 2. HAE Dong Studyroom reservation system

4.2 HAE Dong Academic Information Room

HAE Dong Academic Information Room is a facility free to use for all students who belong to College of Information and Communication Engineering (CICE). It is divided into four spaces which are Information Lounge, Seminar Rooms, Information Research Room, and Study Room. Each space has a different purpose for students to use in need. However, there exists no reservation system for all spaces but the seminar rooms. Therefore, difficulty exists for students to check available seats in each space before visiting.

4.3 SKKU Campus Map

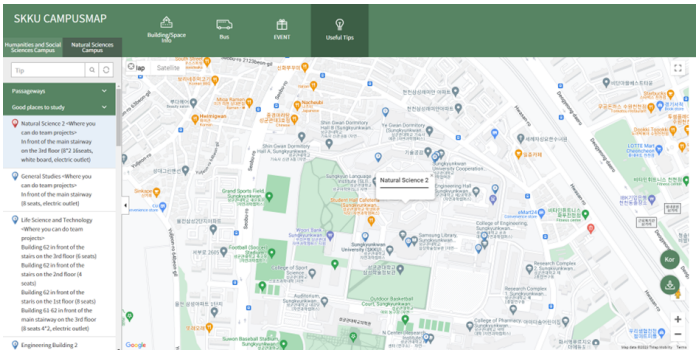

Fig. 3. SKKU Campus Map

SKKU Campus Map is a campus map which includes information about the whole campus. In the service, information about each building, facility, space,

bus stops and even useful tips such as passageways or places to study are provided. For students, it introduces useful information one might not have if not familiar with the campus. Even though it introduces good places to study on campus with total number of seats in each space, it does not include the number of empty seats in real-time. Therefore, students may experience inconvenience visiting themselves to find if the space is available.

5 Background

5.1 YOLO5

In the domain of object detection, the YOLO (You Only Look Once) framework presents a distinctive methodology. Rather than the conventional approach, which often involves scanning images on a pixel-by-pixel basis, YOLO employs a single convolutional network to detect the entirety of an image in a singular forward pass. This framework partitions the image into a structured grid where each segment undergoes classification and localization to define objects and their inherent structures. Uniquely, YOLO utilizes a regression-based algorithm for predicting bounding boxes, deviating from traditional classification-based methods. In the standard approach, an initial Region Of Interest (ROI) is identified, which is then subjected to a Convolutional Neural Network (CNN) for object detection. The efficiency of YOLO's regression algorithm lies in its capability to predict bounding boxes for the entire image simultaneously, culminating in significantly accelerated processing speeds.

5.2 OpenCV

OpenCV stands for Open Source Computer Vision and is an open-source library suitable for various image/video processing tasks. Distributed under the BSD license, OpenCV is free for both academic and commercial purposes. This means businesses and individual developers can create products using OpenCV without having to share the source code or pay licensing fees, making it a popular choice for many. Additionally, it supports various interfaces including C++, C, Python, and Java, and is compatible with multiple operating systems like Windows, Linux, Mac OS, iOS, and Android. OpenCV is designed with an emphasis on computational efficiency and real-time applications. As a result, even by simply utilizing the APIs provided by OpenCV, one can develop applications capable of real-time processing without deep considerations of optimization or algorithms. This allows for the creation of high-quality commercial software without extensive algorithmic knowledge. Furthermore, OpenCV supports multi-core processing, making it adaptable to a variety of situations.
The adoption of YOLO and OpenCV within this project is grounded in their distinctive capabilities designed for advanced image processing tasks. Specifically,

YOLO was chosen for its exceptional processing speed and its proficiency in detecting objects across an entire image in a singular forward pass. This capacity is paramount, especially in dynamic environments demanding rapid adaptation to fluctuating scenarios. OpenCV's selection is rooted in its exceptional versatility and its extensive support for myriad use-cases. Its compatibility with a diverse array of interfaces and operating systems facilitates its application across various platforms. Furthermore, OpenCV's licensing structure, which accommodates both commercial and academic endeavors, enhances the potential applicability of our project across multiple domains.

6 Architecture

7 Proposed Solution (Implementation)

7.1 AI Implementation

Our sophisticated AI solution harnesses the synergies of computer vision and machine learning to offer precise real-time seating availability detection.

Detailed Workflow:

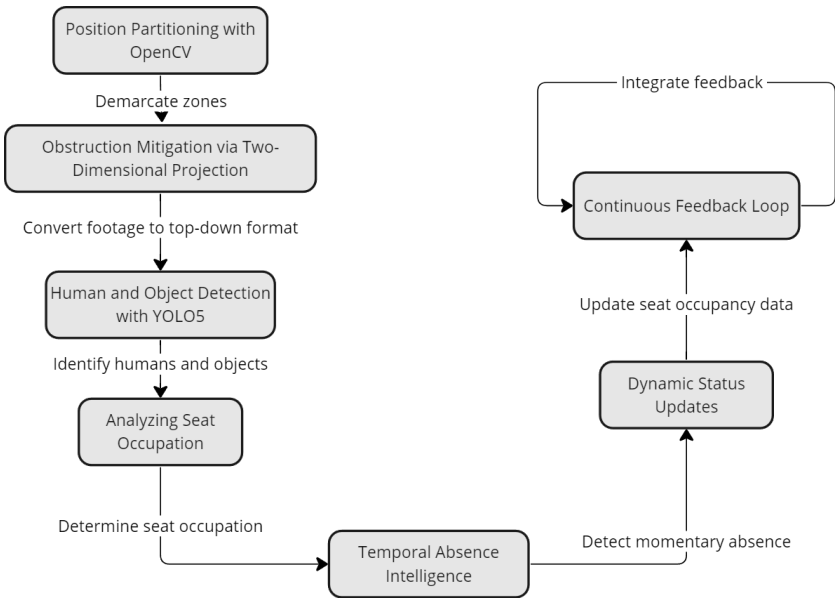

Fig. 4. AI used Data Process Flow

1. **Position Partitioning with OpenCV**: The initial step involves demarcating specific zones within the video feed using OpenCV. Each of these zones corresponds to a potential seating or usable space.

2. **Obstruction Mitigation via Two-Dimensional Projection**: A single-camera setup, albeit efficient, has its limitations. Objects or individuals closer to the camera can potentially obscure the view of seats located further back. To address this, we're exploring the integration of a two-dimensional projection mechanism. By converting and synthesizing the camera's footage into a top-down or sky-view format, obstructions are significantly reduced, offering a more comprehensive visibility of all seats.

3. **Human and Object Detection with YOLO5**: The YOLO5 model, known for its superior object detection capabilities, plays a pivotal role. It is tasked with identifying not just humans but also objects within the predefined zones. Given its optimization for real-time scenarios, it ensures immediate and accurate detection.

4. **Analyzing Seat Occupation**: The system's intelligence lies in its ability to discern between transient and prolonged presence. If an entity, be it human or an object like a bag or book, remains in a zone for a duration exceeding a predefined threshold, the seat is considered occupied.

5. **Temporal Absence Intelligence**: Our solution takes into account real-world scenarios. In places like libraries or study areas, individuals often leave their belongings behind momentarily. The system detects such objects and, using a timer mechanism, waits for a certain period before changing the seat's status, thereby preventing false negatives.

6. **Dynamic Status Updates**: Upon determining a seat's occupancy status, the system rapidly updates this data, converting it into a structured JSON format. This real-time data transformation ensures the backend processes the most recent information, and the frontend displays the most accurate seating availability to users.

7. **Continuous Feedback Loop**: The integration of feedback mechanisms, both from user input and system logs, helps in the continuous improvement of detection accuracy. This iterative approach ensures the system evolves and adapts to changing scenarios and user behaviors.

With this holistic AI-driven framework, we aim to offer not just a tool that detects seat availability but an adaptive solution that understands and responds to the intricacies of real-world environments.

7.2 Frontend

The frontend platform, leveraging the power of React.js, is designed to provide an intuitive and dynamic user experience. React.js, with its renowned efficiency and scalability, is the cornerstone of our frontend development.

Key Features:

Component-Based Architecture:
- **Description**: React's architecture is inherently component-based. Each UI piece is a reusable component, allowing for a clean and organized codebase.

- **Method**: We'll start by designing atomic components (buttons, cards, etc.) and gradually compose them to form complex UI structures.

- **Implementation Sequence**:
1. Define atomic components.
2. Develop container components.
3. Integrate atomic components within containers.
4. Implement component state and props for dynamic rendering.

Responsive Design:
- **Description**: In today's diverse device landscape, responsiveness is paramount. Our design will ensure that the interface is fluid across device sizes.

- **Method**: Utilize CSS frameworks like Bootstrap or Flexbox combined with media queries.

- **Implementation Sequence**:
1. Define style guidelines.
2. Implement base styles for mobile views.
3. Use media queries to adapt and optimize for tablet and desktop views.

State Management:
- **Description**: As the application scales, managing state becomes crucial. Tools like Redux will ensure a single source of truth for the application's state.

- **Method**: Implement Redux for global state management, complemented by React's local state for component-specific data.

- **Implementation Sequence**:
1. Define the initial state and actions.
2. Develop reducers to handle state transitions.
3. Integrate Redux store with React components.

4. Implement asynchronous actions with middleware like Redux-Thunk for API calls.

Routing:

- **Description**: For a seamless user experience, we'll implement routing to navigate between different views without traditional page reloads.

- **Method**: Utilize React Router for defining and managing routes.

The above methodologies and sequences provide a robust roadmap for implementing the frontend of the project. Our approach ensures a balance between user experience and code maintainability, setting the stage for a successful project delivery.

7.3 Backend

Our backend infrastructure, anchored by Node.js and augmented with Express.js, is meticulously designed to manage the dynamic and real-time demands of the project.

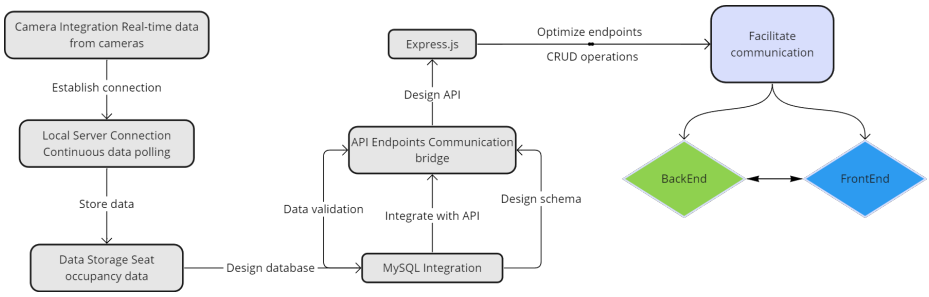

Fig. 5. Backend Data Process Flow

Detailed Workflow:

1. **Camera Integration**:

- **Description**: The system's core functionality revolves around obtaining real-time data from cameras installed at various locations.

- **Method**: Implement a continuous data polling mechanism to fetch data from the local server where the camera is stationed.

- **Implementation Sequence**:
    1. Establish a stable connection with the local server.
    2. Define polling intervals based on real-time requirement specifics.
    3. Implement error-handling mechanisms for potential camera disconnections or malfunctions.

2. **Data Storage**:

- **Description**: Ensuring the swift and accurate storage of seat occupancy data is paramount.

- **Method**: Use MySQL, leveraging its transactional capabilities to ensure data consistency.

- **Implementation Sequence**:
    1. Design a normalized database schema, prioritizing data integrity.
    2. Implement data validation before insertion to ensure only structured JSON is stored.
    3. Regularly backup data to prevent potential data loss scenarios.

3. **API Endpoints with Express.js**:

- **Description**: To facilitate communication between the frontend and backend.

- **Method**: Create RESTful API endpoints.

- **Implementation Sequence**:
    1. Design the API contract (endpoints, request-response structure).
    2. Implement CRUD operations for seat management.
    3. Optimize endpoints for performance using techniques like caching.

4. **Security**:

- **Description**: In today's digital age, security is not just essential, it's paramount.

- **Method**: Implement a multi-layered security approach.

5. **Scalability**:

- **Description**: The solution is designed for growth, ensuring it can handle increased demand.

- **Method**: Leverage the asynchronous capabilities of Node.js and Express.js.

- **Implementation Sequence**:
  1. Optimize the codebase for non-blocking operations.
  2. Monitor server loads and optimize as necessary.
  3. Consider horizontal scaling strategies, such as load balancing, for future expansion.

Our backend is not just a data processor; it's the engine driving real-time updates, ensuring users are always equipped with the most recent and accurate information on seat availability.

8 Planning in Detail

Role Distribution

The project is largely divided into two categories: AI and web development. AI includes data collection, data preprocessing and model/algorithmic implementation. Web development can also be divided into two parts which is front-end and back-end. The roles have been distributed except for the model and algorithmic implementation, which all members will be working together. Although each has their own part, it will be flexible if needed during the project.

Name	Role
Doyeop Kim	Data collection, AI, Back-end
Jaeyoon Park	Data collection, AI, Front-end
Dayeon Woo	Data preprocessing, AI, Front-end
Jimin Choi	Data preprocessing, AI, Front-end

Development Plan

Weeks	2 3	4 5	6 7	8 9	10 11	12 13	14 15
Refinement of Topic	O						
Data collection	O	O	O	O			
AI Model Design		O	O	O			
Data Preprocessing		O	O	O			
Model & Algorithm Implementation			O	O			
UI/UX Design & Implementation				O	O	O	
Beta Service Launch						O	O
Incorporation of Feedback & Revision						O	O

The project is planned largely to four parts: data collection and preprocessing, AI, web development and testing. The schedule includes total of 15 weeks, where each subtask is divided into two weeks based. It follows in the order of main four parts with subtasks included. Although it is the schedule planned, it is aware some might change during the project depending on situation and progress.

References

1. Redmon, J., Divvala, S., Girshick, R., & Farhadi, A. (2016). *You Only Look Once: Unified, Real-Time Object Detection*. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (pp. 779-788).
2. Bradski, G., & Kaehler, A. (2008). *Learning OpenCV: Computer vision with the OpenCV library*. O'Reilly Media, Inc.
3. **SKKU CAMPUSMAP**. *SKKU CAMPUSMAP*.
   Available at: https://eng.skku.edu/eng/About/campusinfo/CampusMap.do.
4. **YOLOv5 GitHub Repository** *YOLOv5 GitHub Repository*.
   Available at: https://github.com/ultralytics/yolov5.git.
5. **HAE Dong Studyroom Reservation System**. *HAE Dong Studyroom Reservation System*.
   Available at: https://scg.skku.ac.kr/seminar/.